

University of Wollongong

## Research Online

---

Faculty of Engineering and Information  
Sciences - Papers: Part A

Faculty of Engineering and Information  
Sciences

---

1-1-2015

### Privacy-preserving encryption scheme using DNA parentage test

Clementine Gritti

*University of Wollongong*, [cjpg967@uowmail.edu.au](mailto:cjpg967@uowmail.edu.au)

Willy Susilo

*University of Wollongong*, [wsusilo@uow.edu.au](mailto:wsusilo@uow.edu.au)

Thomas Plantard

*University of Wollongong*, [thomaspl@uow.edu.au](mailto:thomaspl@uow.edu.au)

Khin Than Win

*University of Wollongong*, [win@uow.edu.au](mailto:win@uow.edu.au)

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>



Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

---

#### Recommended Citation

Gritti, Clementine; Susilo, Willy; Plantard, Thomas; and Win, Khin Than, "Privacy-preserving encryption scheme using DNA parentage test" (2015). *Faculty of Engineering and Information Sciences - Papers: Part A*. 4109.

<https://ro.uow.edu.au/eispapers/4109>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: [research-pubs@uow.edu.au](mailto:research-pubs@uow.edu.au)

---

## Privacy-preserving encryption scheme using DNA parentage test

### Abstract

Consider the following practical scenario. Another Alice would like to make her assets accessible only to her descendants. In order to do so, she encrypts her secret Swiss bank account with her DNA sequences, and provides this information to her descendants. To simplify the scenario and without losing generality, we assume that Alice has only one son, named Bob. Therefore, Alice provides the ciphertext to her family (this ciphertext can eventually be published), which will later be stored in a secure cloud storage. Later, when Alice is unable to access her assets herself (due to her illness for instance), then she can ask Bob to use his DNA to conduct a successful decryption. The decryption is successful if and only if Bob is a true descendant of Alice (i.e., Bob passes the DNA parentage test). Furthermore, a third party Charlie will not be able to conduct a successful decryption, even if he colludes with other people who are not Alice's relatives, since Charlie does not have the required DNA sequences, and hence, he will fail the DNA parentage test. Additionally, Charlie will not learn about anything else other than the unsuccessful decryption process.

### Keywords

parentage, test, scheme, encryption, preserving, privacy, dna

### Disciplines

Engineering | Science and Technology Studies

### Publication Details

Gritti, C., Susilo, W., Plantard, T. & Win, K. (2015). Privacy-preserving encryption scheme using DNA parentage test. *Theoretical Computer Science*, 580 1-13.

# Privacy-Preserving Encryption Scheme using DNA Parentage Test

1

Clémentine Gritti, Willy Susilo, Thomas Plantard and Khin Than Win  
Centre for Computer and Information Security Research  
School of Computer Science and Software Engineering  
University of Wollongong, Australia  
Email: {cjpg967@uowmail.edu.au}, {wsusilo, thomaspl, win}@uow.edu.au

## Abstract

Consider a situation where a user's credential is linked with DNA sequences. This credential can be used to construct a confidential message, which is decipherable by someone who is the relative of the encryptor, *i.e.*, the person who has *similar* DNA sequences. We would like to have a system that provides privacy-preserving, in the sense that nobody will learn whether the encryptor is truly the ancestor of the recipient, except the fact whether the recipient can decrypt the given ciphertext or not. We present a system to realize this complex access control on encrypted data, such that the attributes used to describe a user's credential will determine a policy as to which recipient will be able to decrypt the provided ciphertext. To enhance the system, we add an extra feature, namely a token controlled decryption. A ciphertext can be delivered to a receiver, but the ciphertext cannot be decrypted until the token is provided. We provide a realistic scenario to capture this situation. We formalize the security model of our system against chosen ciphertext attacks, malicious adversaries and collusion attacks. Furthermore, we show an efficient and provably secure construction.

## Index Terms

Attribute-Based Encryption, Embedded-Token Public Key Encryption.

## I. INTRODUCTION

Consider the following real life scenario. A millionaire Alice would like to make an encrypted will, which contains her secret Swiss bank account, to be given to her descendents. To simplify the scenario and without losing generality, we assume that Alice has only one son, named Bob. When making her will, Alice constructs a ciphertext (containing the secret Swiss bank account), which is a message encrypted with Alice's DNA sequences. Then, Alice provides the ciphertext to her family (this ciphertext can eventually be published), which will later be stored in a secure cloud storage. Additionally, Alice creates a secret "token", which is given to her lawyer. After Alice passes away, Bob could first retrieve the ciphertext. Additionally, Bob will require the secret token from the lawyer to allow a successful decryption. Finally, Bob will need to use his DNA to conduct a successful decryption. The decryption is successful if and only if Bob is a real direct descendent of Alice (*i.e.*, Bob passes the DNA parentage test). One may wonder why the secret token is required in this scenario. This is to stop Bob to conduct an early decryption, *i.e.*, the case where Alice is still alive, but Bob just wants to take Alice's money away. Since the lawyer has been sworn to keep the secret token until the time when Alice passes away, then Bob will not be able to conduct a successful decryption, even if he satisfies all other criterias required. Furthermore, a third party Charlie will not be able to conduct a successful decryption, even if he collides with the lawyer, since Charlie does not have the required DNA sequences, and hence, he will fail the DNA parentage test. We note that this also captures the situation where the lawyer is dishonest and would like to decrypt the ciphertext himself/herself. Note that in this situation, Charlie will not learn about anything else other than the unsuccessful decryption process.

### A. DNA Parentage Test [18]

DNA Parental test is currently the most advanced and accurate technology to determine parental relationship. It uses genetic fingerprints to determine whether two individuals have a biological parent-child relationship. Hence, it establishes genetic evidence whether a man/woman is the biological father/mother of an individual. When testing parental relationship, the result (called the "probability of parentage") is 0 when the parent is not biologically related to the child, and it is typically closed to 1 (but is not necessarily equal to 1) otherwise.

Some but rare individuals are called "chimeras", due to the fact that they have at least two different sets of genes. Nevertheless, almost all individuals have a single and distinct set of genes. Therefore, we met several

cases of DNA profiling that falsely “proved” that a mother was unrelated to her children. Since this has a negligible probability in theory, we assume that these exceptions are not applicable in our protocol.

In the following, we give some useful definitions to explain the genetics vocabulary that will be used throughout this paper.

**Definition 1 (DNA)** Deoxyribonucleic acid (DNA) is a molecule that encodes the genetic information found in all human organisms, as a sequence of nucleotides using letters G, A, T, and C, corresponding to guanine, adenine, thymine, and cytosine, respectively. The molecule is double-stranded helices, consisting of two long polymers of simple units called nucleotides (molecules with backbones made of alternating sugars and phosphate groups). These characters allow the DNA to be well-suited for biological information storage.

**Definition 2 (Chromosome)** A chromosome is an organized structure of DNA found in cells. This single piece of coiled DNA contains many genes, regulatory elements and other nucleotide sequences. It also holds DNA-bound proteins, which serve to package the DNA and control its functions.

**Definition 3 (Allele)** An allele is the alternative forms of the same gene for a character, producing different effects. For instance, different alleles can result in different observable phenotypic traits, such as different pigmentation. However, many variations at the genetic level result in little or no observable variation.

**Definition 4 (Minisatellites)** Minisatellites (or VNTR for Variable Number Tandem Repeat) are repeated combined sequences, such that the size of one sequence is 10 to 60 nucleotides. They are present in all species and are particularly studied in humans, and largely found in the genome. We can easily observe replication errors in these minisatellites, including replication slippage, which are at the origin of interindividual on the number of repetition variations. This variability has many applications, as for DNA Parentage test.

*Principle of DNA Parentage Test*

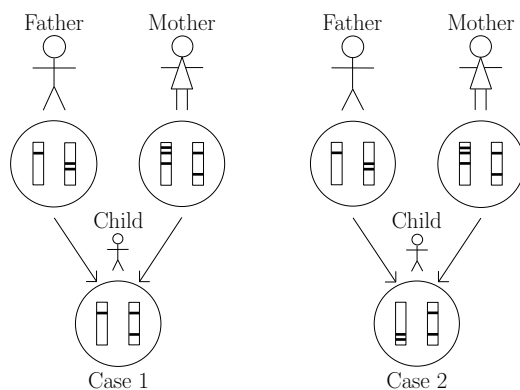


Figure 1. DNA Parentage Test. In Case 1, the father and the mother have biological parent-child relationship with the child since the child’s chromosomes matches one chromosome of each parents’ pair. However, in Case 2, the father is not the biological father of the child since they do not share any common chromosome.

Each individual has in its chromosomes portions of DNA that encode genes and other for which no usefulness has yet been discovered. This last part has an interesting characteristic: some sequences of nucleotide pairs follow on from each other, identically repeated. These repeated sequences are called *minisatellites*. The size of these minisatellites, corresponding to the number of repetitions of DNA sequences, highly varies from person to person, comprised from 5 to 55.

This size is a characteristic of the chromosome. Therefore, it inherits in the same way that its alleles: for each chromosome pair of an individual, there will be one that will match the characteristics of the father and another to those of the mother.

Thus, parental testing is to assess the size of some specific minisatellites. For each chromosome, we need to compare these characteristics between the chromosomes of the child and those of the prospective parents. Then, we can reach some conclusions. To illustrate this idea, please refer to the example below and Figure 1.

*Example.* Let us consider the following example. The following table contains genetic maps of three people: the mother Alice and two presumed sons, labelled as Buz and Charles, respectively. In the table, the size of minisatellite ACGCC is indicated for each chromosome pair. For instance, Alice has the sequence ACGCCACGCCACGCCACGCCACGCCACGCC (number of repetitions = 7) on its first chromosome 14.

Analyzed chrom. pair	Size of the minisatellite		
	Alice	Buz	Charles
Chrom. pair 1	18, 15	5, 15	18, 13
Chrom. pair 8	13, 14	17, 20	55, 14
Chrom. pair 13	34, 15	34, 9	34, 14
Chrom. pair 14	7, 24	5, 14	12, 7

We note that for each chromosome pair, Alice and Charles have one element in common. However, Alice and Buz have only one element in common for the chromosome pair number 1 and the chromosome pair number 13, thus the Buz is not Alice’s biological son. Moreover, we can conclude with a high probability (included in  $[0,1)$ ) that Charles is the biological son of Alice.

### B. Related work

In 1984, the seminal notion of Identity-Based Encryption (IBE) schemes was introduced by Shamir [20]. In these schemes, an authority distributes keys to users with associated identities, and messages are encrypted directly to identities. Shamir did not provide any concrete construction of IBE in [20]. Subsequently, Boneh-Franklin [6] gave the first construction, which was proven secure in the random oracle model. Afterwards, several improvements of this construction have been proposed in the literature. Selectively secure schemes in the standard model were constructed ([7],[4]), then fully secure IBE schemes in the standard model were proposed [5],[21]. The functionality of IBE has also been expanded, which includes a hierarchical structure on identities, where identities can delegate secret keys to their subordinate identities [12],[14]. This particular scheme is called a Hierarchical IBE (HIBE). Moreover, other techniques have been proposed to enhance the security. The dual system encryption methodology was leveraged to obtain fully secure IBE and HIBE systems from simple assumptions [22] and then, extended to obtain a fully secure HIBE system with constant size ciphertexts [15].

In 2005, Sahai and Waters [19] introduced the concept of Attribute-Based Encryption (ABE). Inspired by their scheme, two important but different constructions emerged, namely Ciphertext-Policy Attribute-Based Encryption (the keys are associated with the sets of attributes and the ciphertexts are associated with the access policies) and Key-Policy Attribute-Based Encryption (a ciphertext is related to a set of attributes and each private key corresponds to an access policy over the attributes). Goyal et al. [13] proposed the first KP-ABE. In an orthogonal direction, Bethencourt et al. [3] proposed the first CP-ABE. Later on, the first fully expressive CP-ABE scheme was proposed by Waters [23]. As for IBE, using the dual system encryption methodology, Lewko et al. [16] proposed a stronger secure CP-ABE which leads to some loss of efficiency compared to the most efficient scheme proposed by Waters [23].

In 2005, Baek et al. [1] introduced a new public-key encryption (PKE) scheme, called Token-Controlled PKE (TC-PKE), where individual messages can be encrypted and sent to every receiver, but the latter cannot decrypt the message until he/she is given an extra piece of information, known as the *token*. Later on, Galindo and Herranz [11] added a new security property to thwart that the same ciphertext could decrypt to different messages under different tokens. In 2007, Chow [9] constructed a TC-PKE in the standard model.

*Trivial Solution that will not work.* At the first sight, one may think to incorporate the notion of attribute-based encryption (ABE) schemes, where the characteristic information elements related to Alice’s DNA sequences can be viewed as attributes. Unfortunately, this method will not work due to the following. First, we note that the ABE schemes will not stop Bob from decrypting the ciphertext if he satisfies the policy (*i.e.*, the DNA sequences), and therefore, this will not solve the scenario above. Second, the attributes used to produce the ciphertext in the encryption phase are not exactly the same as the ones used to decrypt the ciphertext. Indeed, if Bob is Alice’s son, then Bob only has half of Alice’s DNA, since the other half comes from Alice’s husband. Therefore, we need to establish a method to verify which attributes are in common between Alice and Bob in order to allow the decryption to proceed, and this task is not a trivial one.

### C. Our Contributions

Motivated by the earlier scenario, we are interested to provide a solution to enable the mechanism to encrypt data using the DNA information of the sender. Then, we allow the receiver to use his DNA information to decrypt the ciphertext, if and only if his DNA sequences *match* the original DNA information used to construct the ciphertext. The matching criteria is done via the parentage test, since DNA has an interesting and useful characteristic, namely the number of repetitions is specific of each individual. These minisatellites’ size of each pair of chromosomes will be treated as attributes and they are used in the encryption/decryption process.

Furthermore, the solution is privacy-preserving, which means that at the end of the protocol no party will learn about anything else with regards to the DNA information.

We provide a new cryptographic primitive called *Token-Controlled Ciphertext-Policy Attribute-Based Encryption* (TC-CP-ABE). We bring the security model for this primitive, and afterwards propose a scheme that satisfies the model. We prove that our scheme is selectively IND-CCA secure under the Decisional Parallel Bilinear Diffie-Hellman Exponent assumption, secure against malicious adversaries and collusion resistant. We need to highlight that the requirement of collusion resistance is hard to achieve. Our scheme is inspired by Waters' ABE scheme, a set intersection protocol and a token-controlled protocol. Nevertheless, we need to stress that a simple and trivial combination of the existing schemes simply will not work, and hence insecure.

#### D. Organization of the Paper

In the next section, we will review some notations that will be used in our definition of the TC-CP-ABE scheme. In section III, we will concentrate on formulating the definition of our scheme. We will present our new construction in Section IV as well as its security and efficiency analysis. Finally, we conclude the paper in Section V.

## II. CRYPTOGRAPHIC TOOLS

In the following, we provide some formal definitions of the cryptographic tools that will be used throughout this paper.

#### A. Access Structure [2]

**Definition 5** Let  $\{P_1, \dots, P_n\}$  be a set of parties. A collection  $\mathbb{AS} \subseteq 2^{\{P_1, \dots, P_n\}}$  is monotone if the following implication is satisfied for all  $B, C$ :  $B \in \mathbb{AS} \wedge B \subseteq C \Rightarrow C \in \mathbb{AS}$ . An (monotone) access structure is a (monotone) collection  $\mathbb{AS} \subseteq 2^{\{P_1, \dots, P_n\}} \setminus \{\emptyset\}$ . We define the sets in  $\mathbb{AS}$  as the *authorized sets* (we say also that these set satisfy  $\mathbb{AS}$ ) and the sets not in  $\mathbb{AS}$  as the *unauthorized sets*.

In our context, the role of the parties is taken by the attributes. Thus, the access structure  $\mathbb{AS}$  will contain the authorized sets of attributes.

#### B. Linear Secret-Sharing Scheme (LSSS) [2]

**Definition 6** Let  $P_i$  be a secret-sharing scheme over a set of parties  $P$ .  $P_i$  is said *linear* over  $\mathbb{Z}_p$  if:

- 1) The shares for each party form a vector over  $\mathbb{Z}_p$ .
- 2) There is a  $l \times n$  matrix  $M$  (the share-generating matrix for  $\Pi$ ). For all  $i = 1, \dots, l$ , the  $i$ -th row of  $M$ , written as  $M_i$ , is labeled by a party  $\rho(i)$ , where  $\rho: \{1, \dots, l\} \rightarrow P$ . Let  $v = (s, r_2, \dots, r_n)$  be a column vector, where  $s \in \mathbb{Z}_p$  is the secret to be shared, and  $r_2, \dots, r_n \in \mathbb{Z}_p$  be randomly chosen. Thus,  $Mv$  is the vector of  $l$  shares of the secret  $s$  according to  $\Pi$ . The share  $(Mv)_i$  belongs to party  $\rho(i)$ .

A SSS which is linear, satisfies the following *linear reconstruction property*. Let  $\Pi$  be a LSSS for the access structure  $\mathbb{AS}$ ,  $S \in \mathbb{AS}$  be any authorized set (i.e.  $S$  satisfies  $\mathbb{AS}$ ), and  $I = \{i: \rho(i) \in S\} \subset \{1, \dots, l\}$ . Therefore, there exist constants  $\{w_i \in \mathbb{Z}_p\}_{i \in I}$  satisfying the following implication:  $\{\lambda_i\}$  are valid shares of any secret  $s$  according to  $\Pi \Rightarrow \sum_{i \in I} w_i \lambda_i = s$ .

**Convention:** The vector  $(1, 0, \dots, 0)$  is the target vector for any LSSS. Let  $I$  be a set of rows of  $M$ . If  $I$  is an authorized set, then  $(1, 0, \dots, 0)$  is in the span of  $I$ . If  $I$  is an unauthorized set, then  $(1, 0, \dots, 0)$  is not in the span of  $I$ . There is a vector  $w$  such that  $w \cdot (1, 0, \dots, 0) = -1$  and  $\forall i \in I, w \cdot M_i = 0$ .

#### C. Bilinear Map

**Definition 7** Let  $\mathbb{G}$  and  $\mathbb{G}_T$  be two multiplicative cyclic groups of prime order  $p \in \Theta(2^k)$  (where  $k$  is the security parameter). Let  $g$  be a generator of  $\mathbb{G}$  and  $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$  be a bilinear map with the following properties:

- 1) **Bilinearity:**  $\forall u, v \in \mathbb{G}, \forall a, b \in \mathbb{Z}_p, e(u^a, v^b) = e(u, v)^{ab}$ ,
- 2) **Non-degeneracy:**  $e(g, g) \neq 1_{\mathbb{G}_T}$ .

$\mathbb{G}$  is said to be a bilinear group if the group operation in  $\mathbb{G}$  and the bilinear map  $e$  are both efficiently computable. We can easily see that  $e$  is symmetric since  $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$ . Let **BSetup** denote an algorithm that on input the security parameter  $k$ , outputs the parameters  $(p, g, \mathbb{G}, \mathbb{G}_T, e)$  as defined above.

#### D. Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption [23]

Waters [23] introduced the Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption and showed that it is secure in the generic group model. This assumption can be viewed as a generalization of the Bilinear Diffie-Hellman Exponent Assumption.

**Definition 8** Let  $\mathbb{G}$  be a group of prime order  $p \in \Theta(2^k)$ . Let  $a, s, b_1, \dots, b_q \in \mathbb{Z}_p$  be chosen at random and  $g$  be a generator of  $\mathbb{G}$ . If an adversary  $\mathcal{A}$  is given  $\vec{y}$ :

$$\begin{aligned} &g, g^s, g^a, \dots, g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}}, \\ \forall 1 \leq j \leq q, &g^{s \cdot b_j}, g^{a/b_j}, \dots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \dots, g^{a^{2q}/b_j}, \\ &\forall 1 \leq j, k \leq q, k \neq j, g^{a \cdot s \cdot b_k/b_j}, \dots, g^{a^q \cdot s \cdot b_k/b_j}, \end{aligned}$$

it must remain hard to distinguish  $e(g, g)^{a^{q+1}s} \in \mathbb{G}_T$  from a random element in  $\mathbb{G}_T$ .

An algorithm  $\mathcal{B}$  that outputs  $z \in \{0, 1\}$  has advantage  $\epsilon$  in solving Decisional  $q$ -Parallel BDHE in  $\mathbb{G}$  if:

$$|Pr[B(\vec{y}, T = e(g, g)^{a^{q+1}s}) = 0] - Pr[B(\vec{y}, T = R) = 0]| \geq \epsilon$$

The Decisional  $q$ -Parallel BDHE assumption holds if no polytime algorithm has non-negligible advantage in solving the Decisional  $q$ -Parallel BDHE problem.

#### E. Target Collision Resistant Hash Function

Target Collision Resistant (TCR) hash function was introduced by Cramer and Shoup [10].

**Definition 9** A TCR hash function  $H$  guarantees that given a random element  $x$  which is from the valid domain of  $H$ , a PPT adversary  $\mathcal{A}$  cannot find  $y \neq x$  such that  $H(x) = H(y)$ . We let  $Adv_{H, \mathcal{A}}^{TCR} = Pr[(x, y) \leftarrow \mathcal{A}(1^k) : H(x) = H(y), x \neq y, x, y \in DH]$  be the advantage of  $\mathcal{A}$  in successfully finding collisions from a TCR hash function  $H$ , where  $DH$  is the valid input domain of  $H$ ,  $k$  is the security parameter. If a hash function is chosen from a TCR hash function family,  $Adv_{H, \mathcal{A}}^{TCR}$  is negligible.

#### F. Miscellaneous Definitions

Without losing generality, we can extend our scheme to allow grandparents and grandchildren as actors. For simplicity, we assume that we have both the DNA of the grandfather and of the grandmother. From that, we obtain a intersection set of cardinality  $U$  (in order to get enough matching elements for security point of view). Nevertheless, we differentiate the scheme Parent-Child and Grandparent-Grandchild. Indeed, for the former scheme, we construct the system such that grandchildren cannot collude their DNA sets in order to have enough elements matching the DNA set of their grandparent (we only allow children to try to decrypt the ciphertext). However, for the latter scheme, if a child of the *two* parents tries to decrypt the ciphertext, we can easily see that *all* the elements of the child's set match with the elements of the parents' set, that will not be the case for the grandchild.

#### G. Notations

We let  $S = \{S_{P_1}, \dots, S_{P_U}\}$  where  $S_{P_i} = \{x_{i,1}, x_{i,2}\}$  is the set corresponding to the numbers of sequences for the minisatellite  $X_i$  on the parent's chromosome pair  $i$ . In the same way, we define  $S' = \{S_{C_1}, \dots, S_{C_U}\}$  and  $S_{C_i} = \{\hat{x}_{i,1}, \hat{x}_{i,2}\}$  as the set corresponding to the numbers of sequences for the minisatellite  $X_i$  on the child's chromosome pair  $i$ . We recall that  $\forall i \in [U], |S_{P_i} \cap S_{C_i}| \geq 1$ , giving that  $|S \cap S'| \geq U$  for  $|S| = |S'| = 2U$ . We note that the parent can be seen as the encryptor, the child as the decryptor and the lawyer as the proxy (untrusted party) in our scheme. The lawyer has to follow a particular requirement: he/she enables the child to decrypt the ciphertext at the appointed time by forwarding an extra piece of information, known as the "token".

In this paper, we consider the Token-Controlled Ciphertext-Policy Attribute-Based Encryption schemes, and we let the access structure be  $\mathbb{AS} = (x_{1,1} \vee x_{1,2}) \wedge (x_{2,1} \vee x_{2,2}) \wedge \dots \wedge (x_{U,1} \vee x_{U,2})$ . We assume that the access structure is written in good order: we mean that the first pair  $(x_{1,1} \vee x_{1,2})$  corresponds to the first chromosome pair, the second pair  $(x_{2,1} \vee x_{2,2})$  to the second chromosome pair, and so forth. Moreover, for one pair of numbers, the first element is associated to the left chromosome and the second element to the right chromosome. Once the token is released, the child should decrypt the message of this parent using the set  $S \cap S'$  which must satisfy  $\mathbb{AS}$ .

A probabilistic polynomial time algorithm will be named in short as PPT algorithm. If  $Set$  is a non-empty set, then  $x \in_R$  denotes that  $x$  has been randomly and uniformly chosen in  $Set$ . Additionally, if  $\mathbf{Alg}$  is an (PPT) algorithm, then  $x \leftarrow \mathbf{Alg}(\_)$  that  $\mathbf{Alg}$  has been executed on some specified inputs  $\_$  and its (random) output has been assigned to the variable  $x$ . Finally, for a positive integer  $N$ , let  $[N]$  be equal to  $\{1, \dots, N\}$ .

### III. TOKEN-CONTROLLED CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION (TC-CP-ABE)

TC-CP-ABE (See Figure 2) is a new cryptographic primitive where a sender should express how he/she wants to share data in the Encryption algorithm, by providing a predicate. Moreover, the receiver has a secret key associated with his/her credentials or attributes. More precisely, the receiver with credentials  $x$  decrypts the ciphertext encrypted using the predicate  $f$  if  $f(x) = 1$ . In order to get the correct  $x$ , the receiver needs to learn the intersection of his/her credentials set with the one of the sender, but without learning nothing else about the credentials that are not in common.

In addition, TC-CP-ABE scheme is an encryption scheme where the messages are encrypted by a public key together with a secret token, such that the receiver holding the corresponding secret key cannot decrypt without the respective token. The sender has to delegate the token to an untrusted party (eg. a proxy) a priori. Subsequently, the proxy is responsible for releasing the token when some predefined conditions occur.

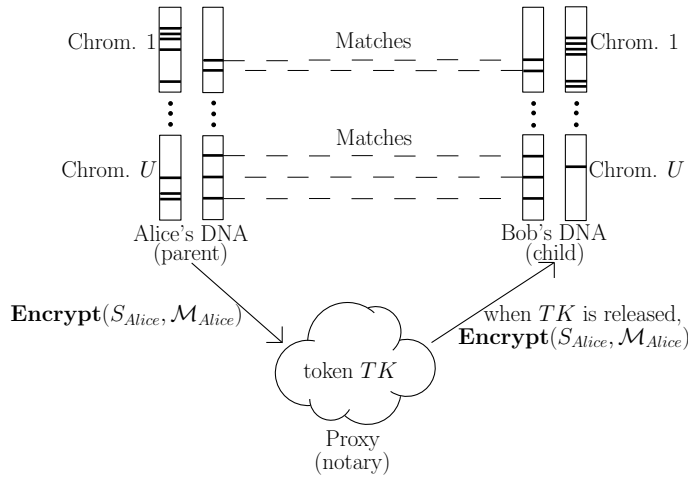


Figure 2. Token-Controlled Ciphertext-Policy Attribute-Based Encryption

#### A. Formal Definition

**Definition 10** A Token-Controlled Ciphertext-Policy Attribute-Based Encryption scheme consists of the following five algorithms:

- 1)  $(PK, MSK) \leftarrow \mathbf{Setup}(k, \mathcal{U})$ : on input a security parameter  $k \in \mathbb{N}$  and an attribute universe  $\mathcal{U}$ , output the public parameters  $PK$  and a master secret key  $MSK$ .
- 2)  $TK \leftarrow \mathbf{TokGen}(k)$ : on input a security parameter  $k \in \mathbb{N}$ , output the token  $TK$ .
- 3)  $SK_S \leftarrow \mathbf{KeyGen}(PK, MSK, S)$ : on input  $PK$ ,  $MSK$  and an attribute set  $S$  that describes the key, output a private key  $SK_S$  for  $S$ . Note that like traditional CP-ABE, each private key  $SK_S$  is associated with an attribute set  $S$ .
- 4)  $CT_{(M, \rho)} \leftarrow \mathbf{Encrypt}(PK, S, (M, \rho), \mathcal{M}, TK)$ : on input  $PK$ , an access structure  $(M, \rho)$  for attributes over  $\mathcal{U}$  and a set of attributes  $S$  satisfying the access structure  $(M, \rho)$ , a plaintext  $\mathcal{M} \in \{0, 1\}^k$  and a token  $TK$ , output a ciphertext  $CT_{(M, \rho)}$ . We assume that the access structure is implicitly included in the ciphertext.
- 5)  $\mathcal{M} \leftarrow \mathbf{Decrypt}(PK, S, SK_S, CT_{(M, \rho)}, TK)$ : on input  $PK$ , an attributes set  $S$  and its corresponding private key  $SK_S$ , a ciphertext  $CT_{(M, \rho)}$  and a token  $TK$ , output a plaintext  $\mathcal{M}$  if  $\mathbb{S}$  satisfies  $(M, \rho)$  where  $\mathbb{S}$  is defined as the intersection set between  $S$  and the attribute set from  $\mathbf{Encrypt}$  algorithm, or a symbol  $\perp$  indicating either  $CT_{(M, \rho)}$  is invalid or  $\mathbb{S}$  does not satisfy  $(M, \rho)$ .

For simplicity, we omit  $PK$  in the expression of the algorithm inputs in the rest of the paper.

*Correctness:* For any  $k \in \mathbb{N}$ , any attribute set  $S$  such that  $S \subseteq \mathcal{U}$ , with its cardinality polynomial to  $k$ , any access structure  $(M, \rho)$  for attributes over  $\mathcal{U}$  and any message  $\mathcal{M} \in \{0, 1\}^k$ , if  $(PK, MSK) \leftarrow \mathbf{Setup}(k, \mathcal{U})$ ,  $TK \leftarrow \mathbf{TokGen}(k)$ ,  $SK_{S'} \leftarrow \mathbf{KeyGen}(MSK, S')$ , for all attribute set  $S'$  used in the system, we have  $\mathbf{Decrypt}(S', SK_{S'}, \mathbf{Encrypt}(S, (M, \rho), \mathcal{M}, TK), TK) = \mathcal{M}$ , where  $S$  satisfies  $(M, \rho)$  and  $\mathbb{S}$  satisfies  $(M, \rho)$  such that  $\mathbb{S} = S \cap S'$ .

In  $\mathbf{KeyGen}$  algorithm, we assume the existence of a central trusted party that knows a secret master key and distributes the secret attribute keys to eligible users.

In  $\mathbf{TokGen}$  algorithm, we assume the existence of an untrusted party (eg. a proxy) that knows the token and releases it to eligible users at the appointed time.



### B. IND-CCA Security Model

In the following, we define the IND-CCA security notions for TC-CP-ABE systems.

**Definition 11** A TC-CP-ABE scheme with is selectively IND-CCA secure if no PPT adversary  $\mathcal{A}$  can win the game below with non-negligible advantage. In the game,  $\mathcal{B}$  is the game challenger,  $k$  and  $\mathcal{U}$  are the security parameter and the attribute universe.

**Init.** The adversary outputs a challenge access structure  $(M^*, \rho^*)$  to the challenger.

**Setup.** The challenger runs the **Setup** algorithm and gives the public parameters  $PK$  to the adversary.

**Phase 1.** The adversary is given access to the following oracles:

- 1) *Private key extraction oracle*  $O_{sk}(S)$ : on input an attribute set  $S$ , the challenger runs  $SK_S \leftarrow \mathbf{KeyGen}(MSK, S)$ .
- 2) *Ciphertext decryption oracle*  $O_d(S, CT_{(M, \rho)}, TK)$ : on input an attribute set  $S$ , a ciphertext  $CT_{(M, \rho)}$  and a token  $TK$ , the challenger returns  $\mathcal{M} \leftarrow \mathbf{Decrypt}(S, SK_S, CT_{(M, \rho)}, TK)$  to the adversary, where  $SK_S \leftarrow \mathbf{KeyGen}(MSK, S)$ ,  $TK \leftarrow \mathbf{TokGen}(k)$  and  $\mathbb{S}$  satisfies  $(M, \rho)$  for  $\mathbb{S}$  defined as the intersection set between  $S$  and the attribute set from **Encrypt** algorithm.
- 3) *Set intersection oracle*  $O_{si}(S, S', CT_{(M, \rho)})$ : on input an attribute sets  $S$  (defining the access structure  $(M, \rho)$  during the encryption), an attribute set  $S'$  and a ciphertext  $CT_{(M, \rho)} \leftarrow \mathbf{Encrypt}(S, (M, \rho), \mathcal{M}, TK)$ , the challenger returns the intersection set  $\mathbb{S} = S \cap S'$  to the adversary, where  $TK \leftarrow \mathbf{TokGen}(k)$ .

Note that if the ciphertexts queries to oracle  $O_d$  are invalid, then the challenger simply outputs  $\perp$ . In this phase, it is forbidden to issue the following query:  $O_{sk}(S)$  for any  $S$  satisfying  $(M^*, \rho^*)$ .

**Challenge.** The adversary submits two equal length messages  $\mathcal{M}_0$  and  $\mathcal{M}_1$ . The challenger flips a random coin  $b$  and encrypts  $\mathcal{M}_b$  under  $(M^*, \rho^*)$ . The ciphertext  $CT_{(M^*, \rho^*)}^*$  is given to the adversary.

**Phase 2.** Phase 1 is repeated except for the following:

- 1)  $O_{sk}(S)$  for any  $S$  satisfying  $(M^*, \rho^*)$ ,
- 2)  $O_d(S, CT_{(M^*, \rho^*)}^*)$  for any  $\mathbb{S}$  satisfying  $(M^*, \rho^*)$  where  $\mathbb{S}$  is defined as the intersection set between  $S$  and the attributes set from **Encrypt** algorithm.

**Guess.** The adversary outputs a guess  $b' \in \{0, 1\}$ . If  $b' = b$  then the adversary wins.

The advantage of  $\mathcal{A}$  is defined as  $\epsilon_1 = Adv_{TC-CP-ABE, \mathcal{A}}^{IND-CCA}(k, \mathcal{U}) = |Pr[b' = b] - 1/2|$ .

### C. Security Model against malicious adversaries

In the following, we define the security notions against malicious adversaries for TC-CP-ABE systems.

**Definition 12** A TC-CP-ABE scheme with is selectively secure against malicious adversaries if no PPT adversary  $\mathcal{A}$  can win the game below with non-negligible advantage. In the game,  $\mathcal{B}$  is the game challenger,  $k$  and  $\mathcal{U}$  are the security parameter and the attribute universe.

**Init.** The adversary outputs a challenge access structure  $(M^*, \rho^*)$  to the challenger.

**Setup.** The challenger runs the **Setup** algorithm and gives the public parameters  $PK$  to the adversary.

**Phase 1.** The adversary is given access to the following oracle:

*Embedded-token encryption oracle*  $O_{et}(\mathcal{M}, TK)$ : on input a message  $\mathcal{M}$  and a token  $TK$ , the challenger returns  $CT_{(M, \rho)} \leftarrow \mathbf{Encrypt}(S, (M, \rho), \mathcal{M}, TK)$  to the adversary where  $S$  satisfies  $(M, \rho)$  and  $TK \leftarrow \mathbf{TokGen}(k)$ .

**Challenge.** The adversary submits two equal length messages  $\mathcal{M}_0$  and  $\mathcal{M}_1$ . The challenger flips a random coin  $b$  and encrypts  $\mathcal{M}_b$  under  $(M^*, \rho^*)$ . The ciphertext  $CT_{(M^*, \rho^*)}^*$  is given to the adversary.

**Phase 2.** Phase 1 is repeated.

**Guess.** The adversary outputs a guess  $b' \in \{0, 1\}$ . If  $b' = b$  then the adversary wins.

The advantage of  $\mathcal{A}$  is defined as  $\epsilon_2 = Adv_{TC-CP-ABE, \mathcal{A}}^{SH}(k, \mathcal{U}) = |Pr[b' = b] - 1/2|$ .

### D. Security Model against collusion attacks

As in previous ABE schemes, one of the main challenges in designing our scheme is to prevent against attacks from colluding users.

The receiver owns a private key which is associated with a attribute set  $S$ . He can decrypt a ciphertext if and only if the access matrix associated with the ciphertext is satisfied by his credentials (or attributes). In order to avoid collusions, each key is randomized with a freshly chosen exponent  $t$ . During the Decryption, each share is multiplied by  $t$  in the exponent. This factor should bind the components of one receiver's key together, therefore we are not be able to combine them with another receiver's key components. Thus, these randomized shares are only useful for one particular key.

In addition, other collusion attacks could occur when constructing the intersection set of attributes  $\mathbb{S}$ : several receivers could combine their sets in order to find enough elements to determine  $\mathbb{S}$  (suppose that some

grandchildren try to decrypt the message of their grandparent). In order to avoid such attacks, a receiver's private key receives extra random pieces of information useful to construct  $\mathbb{S}$  during decryption, using another freshly chosen exponent  $\tilde{t}$ . Intuitively, these pieces should enable one to recover the coefficients of the polynomials constructed from the sender's attributes set and to test whether the elements of a receiver's attributes set are roots of these polynomials.

## IV. OUR CONSTRUCTION

### A. Description

In this section, we present a construction of TC-CP-ABE scheme in the random oracle model with selectively IND-CCA security, security against malicious adversaries and collusion resistance. Inspired by Waters ABE [23], we first construct a selectively IND-CCA secure and collusion resistant ABE scheme in the random oracle model. The main idea is to realize expressive functionality and to prove security under the concrete and non-interactive Decisional Parallel Bilinear Diffie-Hellman Exponent assumption. As in the Waters construction, the encryption algorithm takes as input a LSSS access matrix  $M$  and distributes a random exponent  $s \in \mathbb{Z}_p$  according to  $M$ . Private keys are generated by a central trusted party and randomized to avoid collusion attack.

Afterwards, we extend it to achieve a TC-CP-ABE scheme with the desired level of security. We define our scheme such that the receiver owns a dataset and wishes to perform a set intersection operation with the sender's dataset without learning any extra information other than the output of the operation. This latter requirement is based on the difficulty of the Discrete Logarithm problem. Moreover, in order to enhance the security, the message is encrypted by using the public parameters together with a secret token, randomly generated by a untrusted party, in such a way that the receiver is not able to decrypt the ciphertext until the token is released by this party.

### B. Our TC-CP-ABE Scheme

We let  $\mathcal{U}$  be the attribute universe in the system and  $S$  be an attribute set such that  $S \subseteq \mathcal{U}$  and  $|S| = 2U$  where  $U$  is a large number (the cardinality is assumed to be public).

- **Setup**( $k, \mathcal{U}$ ). Given a security parameter  $k$  and  $\mathcal{U}$ , run  $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathbf{BSetup}(k)$ . Choose 2 random values  $\alpha, a \in_R \mathbb{Z}_p^*$ , a random generator  $g_1 \in_R \mathbb{G}$  and compute  $h = g^\alpha$ ; choose  $U$  random values  $q_1, \dots, q_U \in_R \mathbb{Z}_p^*$ , and compute  $Q_1 = g^{q_1}, \dots, Q_U = g^{q_U}$ ; and set the following TCR hash functions  $H_1 : \{0, 1\}^{2k} \rightarrow \mathbb{Z}_p^*, H_2 : \mathbb{G}_T \times \mathbb{G} \rightarrow \{0, 1\}^{2k}, H_3 : \{0, 1\}^* \rightarrow \mathbb{G}, H_4 : \{0, 1\}^* \rightarrow \mathbb{G}, H_5 : \mathbb{G}_T \rightarrow \mathbb{G}$ . The public parameters are  $PK = (p, g, \mathbb{G}, \mathbb{G}_T, e, g_1, h, e(g, g)^\alpha, Q_1, \dots, Q_U, H_1, H_2, H_3, H_4, H_5)$  and the master secret key is  $MSK = (g^\alpha, q_1, \dots, q_U)$ .
- **TokGen**( $k$ ). Given a security parameter  $k$ , choose a random token  $TK \leftarrow \mathbf{TokGen}(k)$ .
- **KeyGen**( $MSK, S$ ). Given  $MSK$  and an attribute set  $S$ , choose  $t, \tilde{t} \in_R \mathbb{Z}_p^*$ , and set the private key  $SK_S$  as  $SK_S = (K = g^\alpha h^t, L = g^{\tilde{t}}, \forall x \in S K_x = H_3(x)^t, \bar{Q} = g^{\tilde{t}}, \bar{Q}_1 = g^{-q_1} h^{\tilde{t}}, \dots, \bar{Q}_U = g^{-q_U} h^{\tilde{t}})$ .
- **Encrypt**( $S, (M, \rho), \mathcal{M}, TK$ ). Taking an LSSS access structure  $(M, \rho)$  ( $M$  is an  $l \times n$  matrix and the function  $\rho$  associates rows of  $M$  to attributes), an attribute set  $S$  satisfying  $(M, \rho)$ , and a message  $\mathcal{M} \in \{0, 1\}^k$  as input, the encryption algorithm works as follows.

- 1) From the attribute set  $S = \{(x_{1,1}, x_{1,2}) \dots, (x_{U,1}, x_{U,2})\}$ ,
  - pick at random  $N_{j,0}$  and  $N_{j,1}, N_{j,2}$  in  $\mathbb{Z}_p^*$  for  $j \in [U]$  (for  $i = 1, 2$ , the  $N_{j,i}$  are supposed not to be equal each other, and they are not equal to  $x_{j,1}, x_{j,2}$ ),
  - for  $j \in [U]$ , construct the polynomials  $P_j(x) = N_{j,0}(x - x_{j,1})(x - x_{j,2})(x - N_{j,1})(x - N_{j,2}) = \sum_{i=0}^4 \nu_{j,i} x^i$ ,
  - then, for  $j \in [U], i \in \{0, \dots, 4\}$ , compute  $g_{j,i} = g^{\nu_{j,i}} Q_j = g^{\nu_{j,i} + q_j}$  and  $X_{j,i} = H_5(e(g^{\nu_{j,i}}, g)) \oplus g^{\nu_{j,i}}$ .
- 2) Then,
  - choose  $\beta \in_R \{0, 1\}^k$ , set  $s = H_1(\mathcal{M} \parallel \beta)$  and a random vector  $\vec{v} = (s, y_2, \dots, y_n) \in_R \mathbb{Z}_p^n$ , where  $y_2, \dots, y_n \in_R \mathbb{Z}_p^*$ ,
  - for  $i = [l]$ , set  $\lambda_i = v \cdot M_i$ , where  $M_i$  is the vector corresponding to the  $i$ -th row of  $M$ ,
  - choose  $r_1, \dots, r_l \in_R \mathbb{Z}_p^*$ , set  $A_1 = (\mathcal{M} \parallel \beta) \oplus H_2(e(g, g)^{\alpha s}, TK), A_2 = g^s, A_3 = g^s, (B_1 = h^{\lambda_1} H_3(\rho(1))^{-r_1}, C_1 = g^{r_1}), \dots, (B_l = h^{\lambda_l} H_3(\rho(l))^{-r_l}, C_l = g^{r_l}), D = H_4(A_1, A_3, (B_1, C_1), \dots, (B_l, C_l), (M, \rho))^s$ , and output the original ciphertext  $CT_{(M, \rho)} = ((M, \rho), A_1, A_2, A_3, (B_1, C_1), \dots, (B_l, C_l), D, (g_{1,0}, X_{1,0}), (g_{1,1}, X_{1,1}), (g_{1,2}, X_{1,2}), (g_{1,3}, X_{1,3}), (g_{1,4}, X_{1,4}), \dots, (g_{U,0}, X_{U,0}), (g_{U,1}, X_{U,1}), (g_{U,2}, X_{U,2}), (g_{U,3}, X_{U,3}), (g_{U,4}, X_{U,4}))$ .

Note that  $\{\rho(i) : 1 \leq i \leq l\}$  are the attributes used in the access structure  $(M, \rho)$ .

- **Decrypt**( $CT_{(M, \rho)}, S, SK_S, TK$ ). Parse the original ciphertext  $CT_{(M, \rho)}$  as  $((M, \rho), A_1, A_2, A_3, (B_1, C_1), \dots, (B_l, C_l), D, (g_{1,0}, X_{1,0}), \dots, (g_{U,4}, X_{U,4}))$ , and the private key  $SK_S$  (for an attribute set  $S$ ) as  $(K, L, \forall x \in S K_x, \bar{Q}, \bar{Q}_1, \dots, \bar{Q}_U)$ . We let  $S'$  be the attribute set used in the **Encrypt** algorithm.

- 1) Start by setting the intersection set  $\mathbb{S} = S \cap S'$ .
  - If  $|S| > 2U$  then output  $\perp$ ; otherwise proceed.
  - For  $j \in [U]$  and  $i \in \{0, \dots, 4\}$ , compute  $g_{j,i} \bar{Q}_j = g^{\nu_{j,i} + q_j} g^{-q_j} h^{\bar{t}} = g^{\nu_{j,i}} h^{\bar{t}}$ , and then  $Y = \frac{e(g^{\nu_{j,i}} h^{\bar{t}}, g)}{e(\bar{Q}, h)} = \frac{e(g^{\nu_{j,i}} g^{\alpha \bar{t}}, g)}{e(g^{\bar{t}}, g^\alpha)} = e(g^{\nu_{j,i}}, g)$ . Finally, for  $j \in [U]$  and  $i \in \{0, \dots, 4\}$ , output  $H_5(Y) \oplus X_{j,i} = H_5(e(g^{\nu_{j,i}}, g)) \oplus H_5(e(g^{\nu_{j,i}}, g)) \oplus g^{\nu_{j,i}} = g^{\nu_{j,i}}$ .
  - For  $j \in [U]$ , construct  $g^{P_j(x)} = g^{\nu_{j,0}} \cdot (g^{\nu_{j,1}})^x \cdot (g^{\nu_{j,2}})^{x^2} \cdot (g^{\nu_{j,3}})^{x^3} \cdot (g^{\nu_{j,4}})^{x^4}$ .
  - For  $j \in [U]$ , for a pair  $(\hat{x}_{j,1}, \hat{x}_{j,2})$  in  $S$ , compute
    - \*  $g^{P_j(\hat{x}_{j,1})}$ : if  $g^{P_j(\hat{x}_{j,1})} = 1$  then  $\hat{x}_{j,1} \in S'$  and stop; otherwise  $\hat{x}_{j,1} \notin S'$  and proceed,
    - \*  $g^{P_j(\hat{x}_{j,2})}$ : if  $g^{P_j(\hat{x}_{j,2})} = 1$  then  $\hat{x}_{j,2} \in S'$  and stop; otherwise  $\hat{x}_{j,2} \notin S'$  and output  $\perp$ .
  - Keep going this process for all  $j \in [U]$  and learn  $\mathbb{S} = S' \cap S$ . If  $\mathbb{S}$  satisfies  $(M, \rho)$  and  $|\mathbb{S}| \geq U$  s.t.  $\forall j \in [U], (\hat{x}_{j,1} \in \mathbb{S} \vee \hat{x}_{j,2} \in \mathbb{S})$ , then proceed; otherwise output  $\perp$ .
- 2) We recall that  $I = \{i : \rho(i) \in \mathbb{S}\} \subset [l]$  and let  $\{w_i \in \mathbb{Z}_p^*\}_{i \in I}$  be the set of constants such that  $\sum_{i \in I} w_i \lambda_i = s$ .
  - Verify the validity of the ciphertext
$$e(A_2, g_1) \stackrel{?}{=} e(g, A_3) e(A_3, H_4(A_1, A_3, (B_1, C_1), \dots, (B_l, C_l), (M, \rho))) \stackrel{?}{=} e(g_1, D)$$

$$\mathbb{S} \text{ satisfies } (M, \rho) \text{ ?}$$

$$e\left(\prod_{i \in I} B_i^{w_i}, g\right) \stackrel{?}{=} e(A_2, g^\alpha) \prod_{i \in I} e(C_i^{-1}, H_3(\rho(i))^{w_i}) \quad (1)$$
  - If Eq. (1) does not hold, output  $\perp$ . Otherwise, proceed with the next step.
  - Compute  $Z = e(A_2, K) / \prod_{i \in I} (e(B_i, L) \cdot e(C_i, K_{\rho(i)}))^{w_i}$  and  $\mathcal{M} \parallel \beta = H_2(Z, TK) \oplus A_1$ . Output  $\mathcal{M}$  if  $A_3 = g_1^{H_1(\mathcal{M} \parallel \beta)}$ , and output  $\perp$  otherwise.

*Correctness:* In order to check the correctness of the intersection set, let  $S$  be the set of attributes used in the **Encrypt** algorithm and  $S'$  be the set of attributes used in the **KeyGen** algorithm. For  $j \in [U]$ , for a pair  $(\hat{x}_{j,1}, \hat{x}_{j,2}) \in S'$ , if  $g^{P_j(\hat{x}_{j,i})} = 1$  where  $i = 1, 2$ , then  $P_j(\hat{x}_{j,i}) = 0$  which means that  $\hat{x}_{j,i}$  is a root of  $P_j$  and  $\hat{x}_{j,i} = x_{j,i'}$  where  $i' = 1, 2$ ,  $x_{j,i} \in S$ : we conclude that  $\hat{x}_{j,i} \in S$ . Reciprocally, if  $\hat{x}_{j,i} \in S$ , then  $\hat{x}_{j,i} = x_{j,i'}$  and  $P_j(\hat{x}_{j,i}) = 0$ : we conclude that  $g^{P_j(\hat{x}_{j,i})} = 1$ .

Once we retrieve the intersection set  $\mathbb{S} = S \cap S'$ , we define  $I = \{i : \rho(i) \in \mathbb{S}\} \subset [l]$  and  $\{w_i \in \mathbb{Z}_p^*\}_{i \in I}$  as the set of constants such that  $\sum_{i \in I} w_i \lambda_i = s$ . We check the validity of the value  $Z$ .

$$Z = \frac{e(A_2, K)}{\prod_{i \in I} (e(B_i, L) \cdot e(C_i, K_{\rho(i)}))^{w_i}} = \frac{e(g^s, g^\alpha h^t)}{\prod_{i \in I} (e(h^{\lambda_i} H_3(\rho(i))^{-r_i}, g^t) \cdot (g^{r_i}, H_3(\rho(i))^t)^{w_i}} = \frac{e(g^s, g^\alpha g^{\alpha t})}{e(g, g^{\alpha t}) \sum_{i \in I} \lambda_i w_i} = e(g^s, g^\alpha)$$

hence  $H_2(Z, TK) \oplus A_1 = H_2(e(g^s, g^\alpha), TK) \oplus (\mathcal{M} \parallel \beta) \oplus H_2(e(g^s, g^\alpha), TK) = \mathcal{M} \parallel \beta$ .

### C. IND-CCA Security Proof

We start this section by giving an overview of the formal security analysis<sup>1</sup>. Let  $CT_{(M^*, \rho^*)}^* = ((M^*, \rho^*), A_1^*, A_2^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), D^*, (g_{1,0}^*, X_{1,0}^*), \dots, (g_{U,4}^*, X_{U,4}^*))$  be the challenge ciphertext of  $\mathcal{M}_b$ . Let  $\mathcal{A}$  be an adversary that follows the constraints defined in the previous section.  $\mathcal{A}$  will use the Ciphertext decryption oracle  $O_d$  in order to try to get a special advantage in guessing the value of the bit  $b$ . Roughly speaking,  $\mathcal{A}$  might modify the challenge ciphertext, and submit the resulting ciphertext to  $O_d$ . Since  $A_1^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*)$  are bound by  $D^*$  as well as the description of  $(M^*, \rho^*)$ , such a change is noticeable with non-negligible probability from Eq. (1). Indeed, we can view  $D^*$  as a signature for such components. In addition, the integrity of  $A_2^*$  is bound by  $A_3^*$ . If the ciphertext is changed, then Eq. (1) will not hold. Therefore, no special advantage in guessing  $b$  leaks to  $\mathcal{A}$ .

**Theorem 1** Suppose the Decisional  $q$ -Parallel BDHE assumption holds in  $(\mathbb{G}, \mathbb{G}_T)$  and  $H_1, H_2, H_3$  and  $H_4$  are TCR hash functions, our TC-CP-ABE scheme is selectively IND-CCA secure in the random oracle model.

Suppose there is an adversary  $\mathcal{A}$  who can break the IND-CCA security of our scheme. We then construct a reduction algorithm  $\mathcal{B}$  to decide whether  $T$  is either equal to  $e(g, g)^{a^{q+1}s}$  or to a random element in  $\mathbb{G}_T$ . The simulator  $\mathcal{B}$  plays the IND-CCA game with  $\mathcal{A}$  as follows.

$\mathcal{B}$  takes in  $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathbf{BSetup}(k)$  and a  $q$ -parallel BDHE instance  $\bar{y}, T$ , where  $T$  is either equal to  $e(g, g)^{a^{q+1}s}$  or to a random element in  $\mathbb{G}_T$ .

<sup>1</sup>Our security analysis is inspired by [17].

**Init.** The adversary gives the challenge access structure  $(M^*, \rho^*)$  to  $\mathcal{B}$ , where  $M^*$  has  $l^*$  rows and  $n^*$  columns such that  $l^*, n^* \leq q$ .

**Setup.** The simulator chooses at random  $\alpha', \gamma \in_R \mathbb{Z}_p$  and implicitly sets  $g_1 = g^\gamma$  and  $\alpha = \alpha' + a^{q+1}$  by letting  $e(g, g)^\alpha = e(g^\alpha, g^{a^q})e(g, g)^{\alpha'}$  (it can be seen that  $\alpha = \alpha' + a^{q+1}$ , which cannot be computed by  $\mathcal{B}$ ).

Then the simulator chooses the TCR hash functions and the exponents  $q_1, \dots, q_U$  as in the real scheme, and sends the public parameters  $PK = (p, g, \mathbb{G}, \mathbb{G}_T, e, g_1, h = g^\alpha, e(g, g)^\alpha, Q_1 = g^{q_1}, \dots, Q_U = g^{q_U}, H_1, H_2, H_3, H_4, H_5)$  to  $\mathcal{A}$ . We note that the public parameters are identical to those in the real scheme for the adversary. At any time,  $\mathcal{A}$  can adaptively query the random oracles  $H_j$  for  $j \in [5]$ , which are controlled by  $\mathcal{B}$ . The simulator maintains the lists  $H_j^{List}$  for  $j \in [5]$ , which are initially empty, and answers the queries to the random oracles as follows.

- $H_1$ : on receipt of an  $H_1$  query on  $(\mathcal{M}, \beta)$ , if there is a tuple  $(\mathcal{M}, \beta, s) \in H_1^{List}$ ,  $\mathcal{B}$  forwards the predefined value  $s$  to  $\mathcal{A}$ , where  $s \in \mathbb{Z}_p^*$ . Otherwise,  $\mathcal{B}$  sets  $H_1(\mathcal{M}, \beta) = s$ , responds  $s$  to  $\mathcal{A}$  and adds the tuple  $(\mathcal{M}, \beta, s)$  to  $H_1^{List}$  where  $s \in_R \mathbb{Z}_p^*$ .
- $H_2$ : on receipt of an  $H_2$  query on  $R_1 \in \mathbb{G}_T, R_2 \in \mathbb{G}$ , if there is a tuple  $(R_1, R_2, \delta_1) \in H_2^{List}$ ,  $\mathcal{B}$  forwards the predefined value  $\delta_1$  to  $\mathcal{A}$  where  $\delta_1 \in \{0, 1\}^{2k}$ . Otherwise,  $\mathcal{B}$  sets  $H_2(R_1, R_2) = \delta_1$ , responds  $\delta_1$  to  $\mathcal{A}$  and adds the tuple  $(R_1, R_2, \delta_1)$  to  $H_2^{List}$  where  $\delta_1 \in_R \{0, 1\}^{2k}$ .
- $H_3$ : on receipt of an  $H_3$  query on  $x \in \mathcal{U}$ , if there is a tuple  $(x, z_x, \delta_{2,x}) \in H_3^{List}$ ,  $\mathcal{B}$  forwards the predefined value  $\delta_{2,x}$  to  $\mathcal{A}$  where  $z_x \in \mathbb{Z}_p^*, \delta_{2,x} \in \mathbb{G}$ . Otherwise,  $\mathcal{B}$  constructs  $\delta_{2,x}$  as follows. Let  $X$  denote the set of indices  $i$  such that  $\rho^*(i) = x$  where  $1 \leq i \leq l^*$ . Namely,  $X$  contains the indices of rows of matrix  $M^*$  that corresponds to the same attribute  $x$ .  $\mathcal{B}$  chooses  $z_x \in_R \mathbb{Z}_p^*$  and sets

$$\delta_{2,x} = g^{z_x} \prod_{i \in X} g^{aM_{i,1}^*/b_i + a^2M_{i,2}^*/b_i + \dots + a^{n^*}M_{i,n^*}^*/b_i}.$$

If  $X = \emptyset$ ,  $\mathcal{B}$  sets  $\delta_{2,x} = g^{z_x}$ .  $\mathcal{B}$  responds  $\delta_{2,x}$  to  $\mathcal{A}$  and adds the tuple  $(x, z_x, \delta_{2,x})$  to  $H_3^{List}$ .

- $H_4$ : on receipt of an  $H_4$  query on  $(A_1, A_3, (B_1, C_1), \dots, (B_l, C_l), (M, \rho))$ , if there is a tuple  $(A_1, A_3, (B_1, C_1), \dots, (B_l, C_l), (M, \rho), \xi_1, \delta_3) \in H_4^{List}$ ,  $\mathcal{B}$  forwards the predefined value  $\delta_3$  to  $\mathcal{A}$  where  $\xi_1 \in \mathbb{Z}_p^*, \delta_3 \in \mathbb{G}$ . Otherwise,  $\mathcal{B}$  sets  $\delta_3 = g^{\xi_1}$ , responds  $\delta_3$  to  $\mathcal{A}$  and adds the tuple  $(A_1, A_3, (B_1, C_1), \dots, (B_l, C_l), (M, \rho), \xi_1, \delta_3)$  in  $H_4^{List}$  where  $\xi_1 \in_R \mathbb{Z}_p^*$ .
- $H_5$ : on receipt of an  $H_5$  query on  $\delta \in \mathbb{G}_T$ , if there is a tuple  $(\delta, \xi_2)$  in  $H_5^{List}$ ,  $\mathcal{B}$  forwards the predefined value  $\xi_2$  to  $\mathcal{A}$ , where  $\xi_2 \in \mathbb{G}$ . Otherwise,  $\mathcal{B}$  sets  $H_5(\delta) = \xi_2$  to  $\mathcal{A}$  and adds the tuple  $(\delta, \xi_2)$  to  $H_5^{List}$ , where  $\xi_2 \in_R \mathbb{G}$ .

In addition,  $\mathcal{B}$  maintains the list  $SK^{List}$  which is initially empty as follows:  $SK^{List}$  records the tuples  $(S, SK_S)$  which are the results of the queries to  $O_{sk}(S)$ .

**Phase 1.** The simulator answers to  $\mathcal{A}$ 's queries as follows.

- *Private key extraction oracle*  $O_{sk}(S)$ :

$SK_S$  is constructed for an attribute set  $S$  as follows. We suppose that we give a private key for a set  $S$  to  $\mathcal{B}$ , where  $S$  does not satisfy  $M^*$  (if  $S$  satisfies  $M^*$  then the simulator outputs a random bit in  $\{0, 1\}$  and aborts the simulation).

First,  $r \in_R \mathbb{Z}_p$  is randomly chosen. Then  $\mathcal{B}$  finds a vector  $\vec{w} = (w_1, \dots, w_{n^*}) \in \mathbb{Z}_p^{n^*}$  such that  $w_1 = -1$  and for all  $i$  where  $\rho^*(i) \in S$ , we have that  $\vec{w} \cdot M_i^* = 0$ . By the definition of a LSSS we gave in the previous section, such a vector must exist. We note that if such a vector did not exist, then the vector  $(1, 0, \dots, 0)$  would be in the span of  $S$ .

Second,  $\mathcal{B}$  defines implicitly  $t$  as  $r + w_1 a^q + w_2 a^{q-1} + \dots + w_{n^*} a^{q-n^*+1}$ , by setting  $L = g^r \prod_{i=1, \dots, n^*} (g^{a^{q+1-i}})^{w_i} =$

$g^t$ . By definition of  $t$ ,  $h^t = g^{at}$  contains a term of  $g^{-a^{q+1}}$ . This term will cancel out with the unknown term in  $g^\alpha$  when  $K$  is computed. Thus,  $\mathcal{B}$  creates  $K$  as

$$\begin{aligned} K &= g^{\alpha'} g^{ar} \prod_{i=2, \dots, n^*} (g^{a^{q+2-i}})^{w_i} = g^{\alpha'} g^{a^{q+1}} g^{-a^{q+1}} \prod_{i=2, \dots, n^*} (g^{a^{q+2-i}})^{w_i} \\ &= g^\alpha (g^r \prod_{i=1, \dots, n^*} (g^{a^{q+2-i}})^{w_i})^a = g^\alpha L^a = g^\alpha g^{at} = g^\alpha h^t. \end{aligned}$$

Third, the simulator calculates  $K_x \forall x \in S$ . Let  $x \in S$  for which there is no  $i$  such that  $\rho^*(i) = x$ , we compute  $K_x = L^{z_x}$ . Therefore, we obtain that  $K_x = L^{z_x} = \delta_{2,x}^t = H_3(x)^t$ .

Nevertheless, it remains more difficult to compute key components  $K_x$  for attributes  $x \in S$  for which there is one  $i$  such that  $\rho^*(i) = x$  ( $x$  is used in the access structure). We need to check that there are no terms of the form  $g^{a^{q+1}/b_i}$  that cannot be simulated; if we have  $M_i^* \cdot \vec{w} = 0$ , then all of these terms cancel.

We define  $X$  as  $X = \{i, \rho^*(i) = x\}$ . Thus, for this case,  $\mathcal{B}$  creates  $K_x$  as follows:

$$\begin{aligned}
K_x &= L^{zx} \prod_{i \in X} \prod_{j=1, \dots, n^*} \left( g^{\frac{a^j}{b_i} r} \prod_{k=1, \dots, n^*; k \neq j} (g^{a^{q+1+j-k}/b_i})^{w_k} \right)^{M_{i,j}^*} \\
&= L^{zx} \prod_{i \in X} \prod_{j=1, \dots, n^*} \left( g^{\frac{a^j}{b_i} r} \prod_{k=1, \dots, n^*; k \neq j} (g^{a^{q+1+j-k}/b_i})^{w_k} \right)^{M_{i,j}^*} \prod_{i \in X} \prod_{j=1, \dots, n^*} \left( g^{(a^{q+1}/b_i)} \right)^{w_j M_{i,j}^*} \\
&= \left( g^r \prod_{i=1, \dots, n^*} g^{(a^{q+1-i} w_i)} \right)^{z_x} \prod_{i \in X} \prod_{j=1, \dots, n^*} \left( g^{\frac{a^j}{b_i} r} \prod_{k=1, \dots, n^*} (g^{a^{q+1+j-k}/b_i})^{w_k} \right)^{M_{i,j}^*} \\
&= \left( g^r \prod_{i \in X} g^{a \frac{M_{i,1}^*}{b_i} + a^2 \frac{M_{i,2}^*}{b_i} + \dots + a^{n^*} \frac{M_{i,n^*}^*}{b_i}} \right)^\kappa \text{ such that } \kappa = \left( r + \sum_{i=1}^{n^*} w_i a^{q-i+1} \right) \\
&= \delta_{2,x}^{(r+w_1 a^q + \dots + w_{n^*} a^{q-n^*+1})} = \delta_{2,x}^t = H_3(x)^t.
\end{aligned}$$

We recall that if  $S$  is not an authorized set for  $(M^*, \rho^*)$ , then  $w \cdot M_i^* = 0$ .

$$\text{Thus, we have } \prod_{i \in X} \prod_{j=1, \dots, n^*} (g^{a^{q+1}/b_i})^{w_j M_{i,j}^*} = g^{a^{q+1} \left( \sum_{i \in X} \sum_{j=1, \dots, n^*} w_j M_{i,j}^* / b_i \right)} = g^0 = 1.$$

In addition, the simulator chooses  $\tilde{t} \in_R \mathbb{Z}_p^*$ , and constructs  $\bar{Q}$  and  $\bar{Q}_1, \dots, \bar{Q}_U$  as in the real scheme. Finally,  $\mathcal{B}$  adds the tuple  $(S, SK_S)$  to  $SK^{List}$  and returns  $SK_S$  to the adversary.

- *Ciphertext decryption oracle*  $O_d(S, CT_{(M,\rho)}, TK)$ :

The simulator checks whether Eq. (1) holds. If not, then either the ciphertext is invalid or  $\mathbb{S} = S \cap S'$  does not satisfy  $(M, \rho)$  (where  $S'$  is the attributes set used to define  $(M, \rho)$  in the **Encrypt** algorithm), and  $\mathcal{B}$  outputs  $\perp$ . Otherwise,  $\mathcal{B}$  proceeds.

- If  $(S, SK_S) \in SK^{List}$  for any  $S$  such that  $\mathbb{S} = S \cap S'$  satisfying  $(M, \rho)$ , the simulator recovers  $\mathcal{M}$  as in the real scheme using  $SK_S$ .
- Otherwise,  $\mathcal{B}$  verifies whether  $(\mathcal{M}, \beta, s) \in H_1^{List}$  and  $(R_1, R_2, \delta_1) \in H_2^{List}$  such that  $A_3 = g^s$ ,  $A_1 = (\mathcal{M} \parallel \beta) \oplus \delta_1$ ,  $R_1 = e(g, g)^{\alpha s}$  and  $R_2 = TK$ . The simulator outputs  $\perp$  if no such tuples exist and outputs  $\mathcal{M}$  otherwise.

- *Set intersection oracle*  $O_{si}(S, S', CT_{(M,\rho)})$ :

The simulator parses the ciphertext as  $((M, \rho), A_1, A_2, A_3, (B_1, C_1), \dots, (B_l, C_l), D, (g_{1,0}, X_{1,0}), \dots, (g_{U,4}, X_{U,4}))$ .

- If  $S$  does not satisfy  $(M, \rho)$  then the simulator outputs  $\perp$  (where  $S$  is the attribute set used to define  $(M, \rho)$  in the **Encrypt** algorithm).
- Otherwise, the simulator constructs  $\mathbb{S}$  as in the real scheme. If  $\mathbb{S} = S \cap S'$  does not satisfy  $(M, \rho)$  then the simulator outputs  $\perp$ . If  $|\mathbb{S}| \geq U$  then the simulator outputs  $\mathbb{S}$ .

**Challenge.** The challenge ciphertext is constructed as follows. The adversary gives two messages  $\mathcal{M}_0$  and  $\mathcal{M}_1$  to the simulator. The simulator flips a coin  $b$ . For each row  $i$  of  $M^*$ ,  $\mathcal{B}$  sets  $x^* = \rho^*(i)$  and issues an  $H_3$  query on  $x^*$  to obtain the tuple  $(x^*, z_{x^*}, \delta_{2,x^*})$ .

We carefully focus on the simulation of the  $B_i$  values since the terms which must be canceled out are contained in. Nevertheless,  $\mathcal{B}$  can choose the secret splitting in order to cancel out these terms. In other words,  $\mathcal{B}$  chooses at random  $y'_2, \dots, y'_{n^*}$  and shares the secret using the vector  $\vec{v} = (s, sa + y'_2, \dots, sa^{n-1} + y'_{n^*}) \in \mathbb{Z}_p^{n^*}$ . Moreover, values  $r'_1, \dots, r'_l$  are randomly chosen.

For  $i = 1, \dots, n^*$ , let  $R_i$  be  $R_i = \{k \neq i, \rho^*(k) = \rho^*(i)\}$ . Intuitively,  $R_i$  is the set of all other row indices that have the same attribute as row  $i$ . Therefore, we generate the components of the challenge ciphertext as follows:

$$B_i = \delta_{2,x^*}^{-r'_i} \cdot \left( \prod_{j=2, \dots, n^*} h^{M_{i,j}^* y'_j} \right) \cdot g^{sb_i \cdot (-z_{x^*})} \cdot \left( \prod_{k \in R_i} \prod_{j=1, \dots, n^*} (g^{a^j s (b_i/b_k)})^{M_{k,j}^*} \right)^{-1}, C_i = g^{r'_i} g^{sb_i}.$$

Then,  $\mathcal{B}$  chooses  $\beta^* \in_R \{0, 1\}^k$ ,  $TK^* \in_R \mathbb{G}$ ,  $A_1^* \in_R \{0, 1\}^{2k}$  and implicitly sets  $H_2(T \cdot e(g^s, g^{\alpha'}), TK^*) = A_1^* \oplus (\mathcal{M}_b \parallel \beta^*)$ , and finally computes  $A_2^* = g^s$  and  $A_3^* = (g^s)^\gamma$ .

The simulator issues an  $H_4$  query on  $A_1^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), (M^*, \rho^*)$  to obtain  $(A_1^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), (M^*, \rho^*), \xi_1^*, \delta_3^*)$ , and sets  $D^* = (g^s)^{\xi_1^*}$ .

Moreover, the simulator computes the values  $g_{j,i}^*$  as follows. From the attribute set  $S^* = \{(x_{1,1}^*, x_{1,2}^*), \dots, (x_{U,1}^*, x_{U,2}^*)\}$ ,

- 1) pick at random  $N_{j,0}^*$  and  $N_{j,1}^*, N_{j,2}^*$  in  $\mathbb{Z}_p^*$ , for  $j \in [U]$ ,

- 2) for  $j \in [U]$ , construct the polynomials  $P_j^*(x) = N_{j,0}^*(x - x_{j,1}^*)(x - x_{j,2}^*)(x - N_{j,1}^*)(x - N_{j,2}^*) = \sum_{i=0}^4 \nu_{j,i}^* x^i$ ,
- 3) then, for  $j \in [U], i \in \{0, \dots, 4\}$ , compute  $g_{j,i}^* = g^{\nu_{j,i}^*} Q_j$  and issues an  $H_5$  query on  $\delta^* = e(g^{\nu_{j,i}^*}, g)$  to obtain  $(\delta^*, \xi_2^*)$  and defines  $X_{j,i}^* = \xi_2^* \oplus g^{\nu_{j,i}^*}$ .

Finally, the simulator outputs the challenge ciphertext  $CT_{(M^*, \rho^*)}^* = ((M^*, \rho^*), A_1^*, A_2^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), D^*, (g_{1,0}^*, X_{1,0}^*), \dots, (g_{U,4}^*, X_{U,4}^*))$  to the adversary.

If  $T = e(g, g)^{a^{q+1}s}$ ,  $CT_{(M^*, \rho^*)}^*$  is a valid ciphertext. By letting  $H_1(\mathcal{M}_b, \beta^*) = s$  and  $r_i = r'_i + sb_i$ , we can check:

$$\begin{aligned} A_1^* &= A_1^* \oplus (\mathcal{M}_b \parallel \beta^*) \oplus (\mathcal{M}_b \parallel \beta^*) = H_2(\text{Te}(g^s, g^{\alpha'}), TK^*) \oplus (\mathcal{M}_b \parallel \beta^*) = H_2(e(g, g)^{\alpha s}, TK^*) \oplus (\mathcal{M}_b \parallel \beta^*), \\ A_2^* &= g^s, A_3^* = g^{s\gamma} = g_1^s, D^* = (g^s)^{\xi_1^*} = H_4(A_1^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), (M^*, \rho^*))^s, \\ B_i^* &= \delta_{2,x^*}^{-r'_i} \left( \prod_{j=2, \dots, n^*} h^{M_{i,j}^* y'_j} \right) \cdot g^{-z_x^* sb_i} \cdot \left( \prod_{k \in R_i} \prod_{j=1, \dots, n^*} g^{\alpha^j s (b_i/b_k) M_{k,j}^*} \right)^{-1} \\ &= \delta_{2,x^*}^{-r'_i} \left( \prod_{j=2, \dots, n^*} g^{\alpha M_{i,j}^* y'_j} \right) \left( \prod_{j=1, \dots, n^*} g^{\alpha^j s M_{i,j}^*} \right) \left( \prod_{j=1, \dots, n^*} g^{\alpha^j s M_{i,j}^*} \right)^{-1} g^{-z_x^* sb_i} \left( \prod_{k \in R_i} \prod_{j=1, \dots, n^*} g^{\alpha^j s (b_i/b_k) M_{k,j}^*} \right)^{-1} \\ &= \delta_{2,x^*}^{-r'_i} g^{\alpha \lambda_i} \left( \prod_{j=1, \dots, n^*} g^{\alpha^j s M_{i,j}^*} \right)^{-1} \cdot g^{-z_x^* sb_i} \cdot \left( \prod_{k \in R_i} \prod_{j=1, \dots, n^*} g^{\alpha^j s (b_i/b_k) M_{k,j}^*} \right)^{-1} \\ &= g^{\alpha \lambda_i} g^{-r'_i z_x^*} g^{-z_x^* sb_i} \left( \prod_{i \in X} g^{\alpha M_{i,1}^*/b_i + \alpha^2 M_{i,2}^*/b_i + \dots + \alpha^{n^*} M_{i,n^*}^*/b_i} \right)^{-r'_i} \left( \prod_{j=1, \dots, n^*} g^{\alpha^j s M_{i,j}^*} \right)^{-1} \\ &\quad \left( \prod_{k \in R_i} \prod_{j=1, \dots, n^*} g^{\alpha^j s (b_i/b_k) M_{k,j}^*} \right)^{-1} \\ &= g^{\alpha \delta_i} \delta_{2,x^*}^{-r'_i - sb_i} = g^{\alpha \delta_i} \delta_{2,x^*}^{-r_i} = g^{\alpha \delta_i} H_3(x^*)^{-r_i} = g^{\alpha \delta_i} H_3(\rho^*(i))^{-r_i} = h^{\delta_i} H_3(\rho^*(i))^{-r_i}, \\ C_i^* &= g^{r'_i + sb_i} = g^{r_i}. \end{aligned}$$

Nevertheless, if  $T \in \mathbb{G}_T$ , then the challenge ciphertext is independent of the bit  $b$  for  $\mathcal{A}$ .

**Phase 2.** Same as in **Phase 1** but with the constraints defined in the Security Model Section on page 7.

**Guess.** The adversary will eventually output a guess  $b'$  of  $b$ . The simulator then outputs 0 to guess that  $T = e(g, g)^{a^{q+1}s}$  if  $b = b'$ ; otherwise, it outputs 1 to indicate that it believes  $T$  is a random group element in  $\mathbb{G}_T$ .

*Analysis of the simulations of the Random Oracles:* The simulations of the random oracles are perfect except  $H_1$  and  $H_2$ . Let  $H_1^*$  and  $H_2^*$  be the events that  $\mathcal{A}$  has queried before the **Challenge** phase  $(\mathcal{M}_b, \beta^*)$  to  $H_1$  (probability of successful query as  $\text{Adv}_{H_1^*, \mathcal{A}}^{TCR}$ ) and  $(R_1^*, R_2^*) = (e(g, g)^{\alpha s}, TK)$  to  $H_2$  (probability of successful query as  $\text{Adv}_{H_2^*, \mathcal{A}}^{TCR}$ ).

In the simulation of  $O_{sk}$ , the responses to  $\mathcal{A}$  are perfect.

In the simulation of  $O_d$ , it might be possible that the simulator cannot provide a decryption for a valid ciphertext. Suppose the adversary can generate a valid ciphertext without querying  $e(g, g)^{\alpha s}$  to  $H_2$ , where  $s = H_1(\mathcal{M}, \beta)$ . Let *Valid* be the event that the ciphertext is valid, *QueryH1* be the event that the adversary has queried  $(\mathcal{M}, \beta)$  to  $H_1$ , and *QueryH2* be the event that the adversary has queried  $(e(g, g)^{\alpha s}, TK)$  to  $H_2$ . From the simulation,  $\Pr[\text{Valid} \mid \neg \text{QueryH}_2] \leq \Pr[\text{QueryH}_1 \mid \text{QueryH}_2] + \Pr[\text{Valid} \mid \text{QueryH}_1 \wedge \neg \text{QueryH}_2] \leq q_{H_1}/2^{2k} + 1/p$  and  $\Pr[\text{Valid} \mid \neg \text{QueryH}_1] \leq q_{H_2}/2^{2k} + 1/p$ , where  $q_{H_1}, q_{H_2}$  are the maximum numbers of random oracle queries to  $H_1, H_2$ . Let  $\Pr[\text{DecErr}]$  be the probability that the event *Valid*  $(\neg \text{QueryH}_1 \vee \neg \text{QueryH}_2)$  occurs, then  $\Pr[\text{DecErr}] \leq (\frac{q_{H_1} + q_{H_2}}{2^{2k}} + \frac{2}{p})q_d$ , where  $q_d$  is the total number of ciphertext decryption queries.

Let *Bad* denote the event that  $(H_1^* \mid \neg H_2^*) \vee H_2^* \vee \text{DecErr}$ , then

$$\epsilon_1 = |\Pr[b' = b] - 1/2| \leq \frac{1}{2} \Pr[\text{Bad}] = \frac{1}{2} \Pr[(H_1^* \mid \neg H_2^*) \vee H_2^* \vee \text{DecErr}] \leq \frac{1}{2} (\text{Adv}_{H_2^*, \mathcal{A}}^{TCR} + \frac{2q_d}{p} + \frac{q_{H_1} + (q_{H_1} + q_{H_2})q_d}{2^{2k}})$$

$$\text{Therefore, } \text{Adv}_{\mathcal{A}}^{D-q-PBDHE} \geq \frac{1}{q_{H_2}} \text{Adv}_{H_2^*, \mathcal{A}}^{TCR} \geq \frac{1}{q_{H_2}} (2\epsilon_1 - \frac{q_{H_1} + (q_{H_1} + q_{H_2})q_d}{2^{2k}} - \frac{2q_d}{p}).$$

#### D. Security Proof against malicious adversaries

We prove the following theorem:

**Theorem 2** Our TC-CP-ABE scheme is secure against malicious adversaries in the random oracle model.

Suppose there is a malicious adversary  $\mathcal{A}$  who can break the security of our scheme. We then let  $\mathcal{B}$  be the simulator playing the game with  $\mathcal{A}$  as follows:

$\mathcal{B}$  takes in  $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathbf{BSetup}(k)$ .

**Init.** The adversary gives the challenge access structure  $(M^*, \rho^*)$  to the simulator, where  $M^*$  has  $l^*$  rows and  $n^*$  columns such that  $l^*, n^* \leq q$ .

**Setup.** The simulator chooses at random  $\alpha, a \in_R \mathbb{Z}_p$  and  $g_1 \in \mathbb{G}$ , and sets  $g = h^\alpha$ . Then the simulator chooses the TCR hash functions and the exponents  $q_1, \dots, q_U$  as in the real scheme, and sends the public parameters  $PK = (p, g, \mathbb{G}, \mathbb{G}_T, e, g_1, h = g^\alpha, e(g, g)^\alpha, Q_1 = g^{q_1}, \dots, Q_U = g^{q_U}, H_1, H_2, H_3, H_4, H_5)$  to the adversary. The public parameters are identical to those in the real scheme for  $\mathcal{A}$ . At any time,  $\mathcal{A}$  can adaptively query the random oracles  $H_j$  for  $j \in [5]$ , which are controlled by  $\mathcal{B}$ . The simulator maintains the lists  $H_j^{List}$  for  $j \in [5]$ , which are initially empty and answers the queries to the random oracles as explained previously.

**Phase 1.** The simulator answers to  $\mathcal{A}$ 's queries as follows:

*Embedded-token encryption oracle  $O_{et}(\mathcal{M}, TK)$ :*

The ciphertext  $CT_{(M, \rho)}$  is indistinguishable even with the secret key  $SK_S$  as long as the token  $TK$  is unknown. The intuitive meaning of this oracle is to ensure that  $CT_{(M, \rho)}$  would not leak any information about  $TK$  that is useful for distinguishing the ciphertexts.

The simulator constructs  $CT_{(M, \rho)}$  as follows:

- $\mathcal{B}$  verifies whether  $(R_1, R_2, \delta_1) \in H_2^{List}$  such that  $R_1 = e(g, g)^\alpha$  and  $R_2 = TK$ . if such tuples exist then the simulator constructs  $CT_{(M, \rho)}$  as in the real scheme.
- Otherwise, the simulator chooses  $TK \in_R \mathbb{G}$  and  $\beta \in_R \{0, 1\}^k$ , sets  $s = H_1(\mathcal{M} \parallel \beta)$  and a random vector  $\vec{v} = (s, y_2, \dots, y_n) \in_R \mathbb{Z}_p^n$ , where  $y_2, \dots, y_n \in_R \mathbb{Z}_p^*$ , then, for  $i = 1, \dots, l$ , sets  $\lambda_i = v \cdot M_i$ , where  $M_i$  is the vector corresponding to the  $i$ -th row of  $M$ , and finally, chooses  $r_1, \dots, r_l \in_R \mathbb{Z}_p^*$ , sets  $A_1 = (\mathcal{M} \parallel \beta) \oplus H_2(e(g, g)^{\alpha s}, TK)$ ,  $A_2 = g^s$ ,  $A_3 = g_1^s$ ,  $(B_1 = h^{\lambda_1} H_3(\rho(1))^{-r_1}, C_1 = g^{r_1}), \dots, (B_l = h^{\lambda_l} H_3(\rho(l))^{-r_l}, C_l = g^{r_l})$ ,  $D = H_4(A_1, A_3, (B_1, C_1), \dots, (B_l, C_l), (M, \rho)^s)$ , outputs the ciphertext  $CT_{(M, \rho)} = ((M, \rho), A_1, A_2, A_3, (B_1, C_1), \dots, (B_l, C_l), D, (g_{1,0}, X_{1,0}), \dots, (g_{U,4}, X_{U,4}))$  where  $(g_{1,0}, X_{1,0}), \dots, (g_{U,4}, X_{U,4})$  are constructed as in the real scheme.

Finally, the simulator returns  $CT_{(M, \rho)}$  to the adversary.

**Challenge.** We build the challenge ciphertext. The adversary gives two messages  $\mathcal{M}_0$  and  $\mathcal{M}_1$  to the simulator. The simulator flips a coin  $b$ . As in the real scheme,  $\mathcal{B}$  constructs and outputs the challenge ciphertext  $CT_{(M^*, \rho^*)}^* = ((M^*, \rho^*), A_1^*, A_2^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), D^*, (g_{1,0}^*, X_{1,0}^*), \dots, (g_{U,4}^*, X_{U,4}^*))$  to the adversary.

**Phase 2.** Same as in **Phase 1**.

**Guess.** The adversary will eventually output a guess  $b'$  of  $b$ . The simulator then outputs 0 if  $b = b'$ ; otherwise, it outputs 1.

*Analysis of the simulations of the Random Oracles:* The simulations of the random oracles are perfect except  $H_2$  in  $O_{te}$ . Let us consider the challenge ciphertext  $CT_{(M^*, \rho^*)}^*$  that the adversary gets from the challenger. We parse  $CT_{(M^*, \rho^*)}^*$  as  $CT_{(M^*, \rho^*)}^* = ((M^*, \rho^*), A_1^*, A_2^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), D^*, (g_{1,0}^*, X_{1,0}^*), \dots, (g_{U,4}^*, X_{U,4}^*))$  where  $A_1^* = (\mathcal{M} \parallel \beta) \oplus H_2(e(g, g)^{\alpha s}, TK^*)$ .

Let us denote by  $QueryH_2$  the event that the adversary queries  $H_2$  at the point  $(R_1^*, R_2^*) = (e(g, g)^{\alpha s}, TK^*)$ . Since  $H_2$  is modeled as a random oracle, the value  $b$  is independent from the adversary's view as long as  $QueryH_2$  does not hold. Therefore,  $Pr[b = b'] = Pr[b = b' \wedge QueryH_2] + Pr[b = b' \wedge \neg QueryH_2] = Pr[b = b' | QueryH_2] Pr[QueryH_2] + \frac{1}{2} Pr[\neg QueryH_2]$ . Due to the fact that the adversary is in possession of the secret key  $SK_S$  for the attribute set  $S$ , it can recover the value  $R_1^* = e(g, g)^{\alpha s}$  from  $CT_{(M^*, \rho^*)}^*$ . If  $QueryH_2$  holds, the adversary can compute  $\mathcal{M}_b$  from  $CT_{(M^*, \rho^*)}^*$  and it can check if  $CT_{(M^*, \rho^*)}^* = ((M^*, \rho^*), A_1^*, A_2^*, A_3^*, (B_1^*, C_1^*), \dots, (B_{l^*}^*, C_{l^*}^*), D^*, (g_{1,0}^*, X_{1,0}^*), \dots, (g_{U,4}^*, X_{U,4}^*))$ . Therefore,  $Pr[b = b' | QueryH_2] = 1$ , and  $Pr[b = b'] = \frac{1}{2} + \frac{1}{2} Pr[QueryH_2]$ .

Finally, it lacks to compute an upper bound for  $Pr[QueryH_2]$ . Note that the adversary has two ways of evaluating  $H_2$  at  $(R_1, R_2^*)$ :

- by directly querying  $H_2$  at  $(R_1, R_2^*)$ ,
- by querying  $O_{et}$  on the message  $\mathcal{M}$  and the token  $TK^*$ : it can recover  $e(g, g)^{\alpha s}$  using the secret key  $SK_S$  and it computes  $H_2(R_1, R_2^*) \oplus A_1 = \mathcal{M} \parallel \beta$ .

It follows that  $Pr[QueryH_2] \leq \frac{q_{H_2} + q_{et}}{2^k}$ , where  $q_{H_2}$  and  $q_{et}$  are the maximum number of random oracles queries

to  $H_2$  and the total number of embedded-token encryption queries respectively. Finally,  $\epsilon_2 = |2Pr[b=b'] - 1| \leq \frac{q_{H_2} + q_{et}}{2^{2k}}$ .

### E. Efficiency

From the simulation, the running time of the challenger is bound by

$$t' \leq t + O(1)(q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + q_{H_5} + q_{sk} + q_{et} + q_d + q_{si}) + t_e(q_{sk}O(n^2) + q_dO(l) + q_{H_1}(q_d + q_{et})O(1) + q_{si}O(U)) + t_p((q_{re} + q_d)O(l))$$

where  $q_{H_i}$  denotes the maximum number of random oracle queries to  $H_i$  ( $i \in [5]$ ),  $q_{sk}$  denote the total numbers of private key extraction queries,  $q_d$  denote the total number of ciphertext decryption queries,  $q_{et}$  denote the total number of embedded-token encryption queries,  $q_{si}$  denote the total number of set intersection queries,  $t_e$  denotes the running time of an exponentiation in  $\mathbb{G}$ ,  $t_p$  denotes the running time of a pairing  $\mathbb{G}_T$ ,  $t$  is the running time of  $\mathcal{A}$ ,  $l$  is the number of rows of  $M$ , and  $2U$  is the number of attributes in the sets used in the scheme.

## V. CONCLUSION

We introduced a solution to encrypt data using the DNA sequences of the sender and to decrypt data using the DNA sequences of the receiver. Focusing on the principle of the DNA parentage test, we provided a new scheme called Token-Controlled Ciphertext-Policy Attribute-Based Encryption (TC-CP-ABE). Based on Waters ABE scheme, we extended it adding two extra features to allow decryption of the ciphertext, namely set of intersection operation and generation of a token released at the appointed time. We proved that our scheme is selectively IND-CCA secure, secure against malicious adversaries and collusion resistant in the random oracle model, under the Decisional Parallel Bilinear Diffie-Hellman Exponent assumption. As for the future work, it would be interesting to consider the selectively/adaptively IND-CCA security in the standard model and/or under weaker (Decisional) Bilinear Diffie-Hellman (Exponent) assumptions.

## ACKNOWLEDGEMENTS

This work is partially supported by ARC Linkage Project LP12020052. W. Susilo is supported by ARC Future Fellowship FT0991397 and ARC Discovery Project DP130101383.

## REFERENCES

- [1] J. Baek, R. Safavi-Naini and W. Susilo, *Token-Controlled Public-Key Encryption*. In ISPEC 2005, LCNS 3439, pages 386-397, 2005.
- [2] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*. In PhD Thesis, Israel Institute of Technology, 1996.
- [3] J. Bethencourt, A. Sahai and B. Waters, *Ciphertext-Policy Attribute-Based Encryption*. In IEEE Symposium on Security and Privacy, pages 321-334 2007.
- [4] D. Boneh and X. Boyen, *Efficient selective-ID Secure Identity-Based Encryption without Random Oracles*. In EUROCRYPT, pages 223-238, 2004.
- [5] D. Boneh and X. Boyen, *Secure Identity-Based Encryption without Random Oracles*. In CRYPTO, pages 443-459, 2004.
- [6] D. Boneh and M. K. Franklin, *Identity-Based Encryption from the Weil Pairing*. SIAM J. Comput. 32(3): 586-615, 2003.
- [7] R. Canetti, S. Halevi and J. Katz, *A Forward-secure Public-key Encryption Scheme*. In EUROCRYPT, pages 255-271, 2003.
- [8] L. Cheung and C. C. Newport, *Provably Secure Ciphertext Policy ABE*. In ACM Conference on Computer and Communications Security, pages 456-465, 2007.
- [9] S. S.M. Chow, *Token-Controlled Public-Key Encryption in the Standard Model*. In ISC 2007, LCNS 4779, pages 315-322, 2007.
- [10] R. Cramer and V. Shoup, *Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack*. In SIAM J. Comput., 33(1), pages 167-226, 2004.
- [11] C. Galindo and J. Herranz, *A Generic Construction for Token-Controlled Public-Key Encryption*. In FC 2006, LCNS 4107, pages 177-190, 2006.
- [12] C. Gentry and A. Silverberg, *Hierarchical ID-Based Cryptography*. In ASIACRYPT, pages 548-566, 2002.
- [13] V. Goyal, O. Pandey, A. Sahai and B. Waters, *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data*. In ACM Conference on Computer and Communications Security, pages 89-98, 2006.
- [14] J. Horwitz and B. Lynn, *Toward Hierarchical Identity-Based Encryption*. In EUROCRYPT, pages 466-481, 2002.
- [15] A. B. Lewko and B. Waters, *New techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts*. In TCC, 2010.
- [16] A. B. Lewko, T. Okamoto, A. Sahai and B. Waters, *Fully Secure Functional Encryption: Attribute-Based Encryption and (hierarchical) Inner Product Encryption*. In EUROCRYPT, volume 6110 of Lecture Notes in Computer Science, pages 62-91, 2010.
- [17] K. Liang, L. Fang, W. Susilo and Duncan S. Wong, *A Ciphertext-Policy Attribute-Based Proxy Re-Encryption with Chosen-Ciphertext Security*. In The 5-th International Conference on Intelligent Networking and Collaborative Systems (INCoS 2013), 2013.
- [18] R. Ostrowski, *Paternity Indices*. In Forensic Bioinformatics, 2nd Annual Conference, Statistics and DNA Profiling Wright State University, 2003.
- [19] A. Sahai and B. Waters, *Fuzzy Identity-Based Encryption*. In Advances in Cryptology EUROCRYPT 2005, volume 3494 of Lectures Notes in Computer Science, pages 457-473, 2005.



- [20] A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*. In CRYPTO, pages 47-53, 1984.
- [21] B. Waters, *Efficient Identity-Based Encryption without Random Oracles*. In EUROCRYPT, pages 114-127, 2005.
- [22] B. Waters, *Dual system Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions*. In CRYPTO, pages 619-636, 2009.
- [23] B. Waters, *Ciphertext-Policy Attribute-Based Encryption: an Expressive, Efficient, and Provably Secure Realization*. In Lecture Notes in Computer Science, pages 53-70, 2011.